2 mip

NASA TECHNICAL TRANSLATION NASA TT F-15,636

# INPUT-OUTPUT SYSTEM IN BASIC LANGUAGE FOR THE HP 2116B COMPUTER

F. Palutan

Translation of: "Sistema di ingresso-uscita in linguaggio basic per il calcolatore HP2116B", Consiglio Nazionale della Richerche. Lab. di Ricera e tecnologia per lo Studio del Plasma nello Spazio. Rome, Italy. February, 1973, 33 pp.

N74-25734

(NASA-TT-F-15636) INPUT-OUTPUT SYSTEM IN BASIC LANGUAGE FOR THE HP 2116B COMPUTER (Scientific Translation Service) 29 p HC $4.50 CSCL 09B G3/08 Unclas 40675

JUN 1974 RECEIVED NASA STI FACILITY INPUT BRANCH

Reproduced by NATIONAL TECHNICAL INFORMATION SERVICE US Department of Commerce Springfield, VA. 22151

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546 JUNE 1974

STANDARD TITLE PAGE

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA TT F-15,636 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| INPUT-OUTPUT SYSTEM IN BASIC LANGUAGE FOR THE HP 2116B COMPUTER | June, 1974 |
| | 6. Performing Organization Code |

| 7. Author(s) | 8. Performing Organization Report No. |
|---|---|
| Franco Palutan | 10. Work Unit No. |

| 9. Performing Organization Name and Address | 11. Contract or Grant No. |
|---|---|
| SCITRAN<br>Box 5456<br>Santa Barbara, CA 93108 | NASw-2483 |
| | 13. Type of Report and Period Covered<br>Translation |
| 12. Sponsoring Agency Name and Address<br>National Aeronautics and Space Administration<br>Washington, D.C. 20546 | 14. Sponsoring Agency Code |

15. Supplementary Notes

Translation of: "Sistema di ingresso-uscita in linguaggio basic per il calcolatore HP2116B", Consiglio Nazionale della Richerche. Lab. di Ricera e tecnologia per lo Studio del Plasma nello Spazio. Rome, Italy. February, 1973, 33 pp.

16. Abstract

Input output routines in BASIC for use with the HP 2116B computer are presented. The limitations of BASIC to teletype conversational mode were overcome to permit its use with other peripheral equipment. The I/O sequences and deck card structure are detailed.

| 17. Key Words (Selected by Author(s)) | 18. Distribution Statement |
|---|---|
| | Unclassified - Unlimited |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 29 | 4.50 |

# INPUT-OUTPUT SYSTEM IN BASIC LANGUAGE FOR THE HP 2116B COMPUTER

Franco Palutan

## Introduction

The present internal report updates and summarizes the previous two reports [1, 2] concerning the use of the HP2116B computer and the programs available for the call of peripherals using Basic language. /1*

The actual configuration of the calculation system, lacking a disc unit and a card reader, makes the compilation and the execution of programs written in advanced languages such as Fortran or Algol rather complex. We consolidated, therefore, the use of Basic language, which permits a direct interaction between the user and the computer, with considerable savings in time, and great simplicity of use of the system.

The Basic, however, in its standard version permits communication to the computer only using a teletype, not using other peripheral units of the system.

---

*Numbers in the margin indicate pagination in the original foreign text.

We wanted to exploit the advantages of Basic, and overcome its limitations. We determined, therefore, to write new programs for the input-output of information from the peripherals, extending, therefore, the speech capacity of this language.

The efficiency and completeness of interface programs between computer and peripherals (input-output drivers) have been increasing with time; many different versions of Basic have been written, some of them described in the two reports mentioned.

The Basic version presented here differs from the previous version, not only because it contains programs for the graphic tracer Calcomp 565, but also because all other programs have been completely rewritten, altough the calls were left almost unchanged for the sake of compatibility. /2

In particular, we used the "interrupt" system of the computer, to make the input-output operations more efficient; furthermore, the complete input-output system was organized in a flexible arrangement, so as to be able to carry out substitutions and modifications of the program with relative ease.

One can add to this version of Basic the library of programs for fast Fourier transfers (FFT) [3]; the calls for these programs are listed in the present report.

We also repeated here, for the sake of completeness, the calls of programs for data acquisition [2], which permit the interfacing of the computer with various instruments (Laben, Voltmeters and HP counters); these are not inserted in the new Basic version.

We reproduced from report [1] in appendix A the fundamental procedures needed to use the computer; we presented in appendix B some typical programs for the reading of data from magnetic tape; we optimized the execution times through use of the "interrupt" machine system.

## 2. System Configuration

The connection between the central calculation unit and the peripherals is carried out through the input-output system, which includes a "hardware" part, described in the manuals, and a "software" part, which is described in this report. /3

Each peripheral is connected to the computer through an interface card to which one assigns an address depending on its position in the input-output system.

The transfer of information from the central unit to the peripherals and vice versa occurs by means of the interface card, addressed directly by the program. One can insert up to 16 cards in the input-output system.

We listed in Table 1, the peripherals, the interface cards, and the addresses (in octal code), according to the actual configuration of the system. It should be noted that the magnetic tape requires two interface cards, one for the transfer of input and output data, and one for control of the unit; non-standard peripherals (multichannel Laben, voltmeters and counter) are connected using interface cards of general use, capable of parallel transfer to the computer of 32 bit data. Cards with two 16 bit registers permit the input and output of data from various instrumentation.

TABLE 1.

| Peripheral | Interface card | Address (octal) |
|---|---|---|
| Mag tape HP 2020 | Mag. tape data | 10 |
| | Mag. tape control | 11 |
| Fast reader | Photoreader input card | 12 |
| Teletype | TTY input/output | 13 |
| Video HP1300A | D/A converter | 14 |
| Laben 1024 channels | HP data - 32 bit input | 15 |

(Table continued on following page)

TABLE 1. (continued)

| Peripheral | Interface card | Address (octal) |
|---|---|---|
| Voltmeter HP3440A | HP data - 32 bit input | 16 |
| Counter HP5233L | HP data - 32 bit input | 17 |
| Voltmeter HP3440A | HP data - 32 bit input | 20 |
| — | 16 bit input/output duplex reg. | 21 |
| — | 16 bit input/output duplex reg. | 22 |
| Calcomp 565 | Plotter interface | 23 |

## 3. Basic Input-Output System

The software part of the input-output system consists of a series of programs which transfer data from the central unit to the peripherals, and vice versa, and which carry out control operations. /5

These programs were added to the BASIC compiler provided by the manufacturer; to increase memory space, the programs concerning matrix calculations were removed.

The programs were written in absolute Assembler; this means that their location in the memory has to be set and established once and for all, because in this case (as for Fortran programs or for movable Assembler) a control program (BCS) is not available, which loads the programs and places them in the memory, providing at the same time during the execution the references required.

A memory map showing the program location is reported in Figure 1.

It should be noted that the different programs were compiled and tested independently, and that modifications or corrections may be carried out easily without complete restructuring of the system.

We used the "interrupt" system of the machine to write the programs; this permits the computer to continue the execution of a

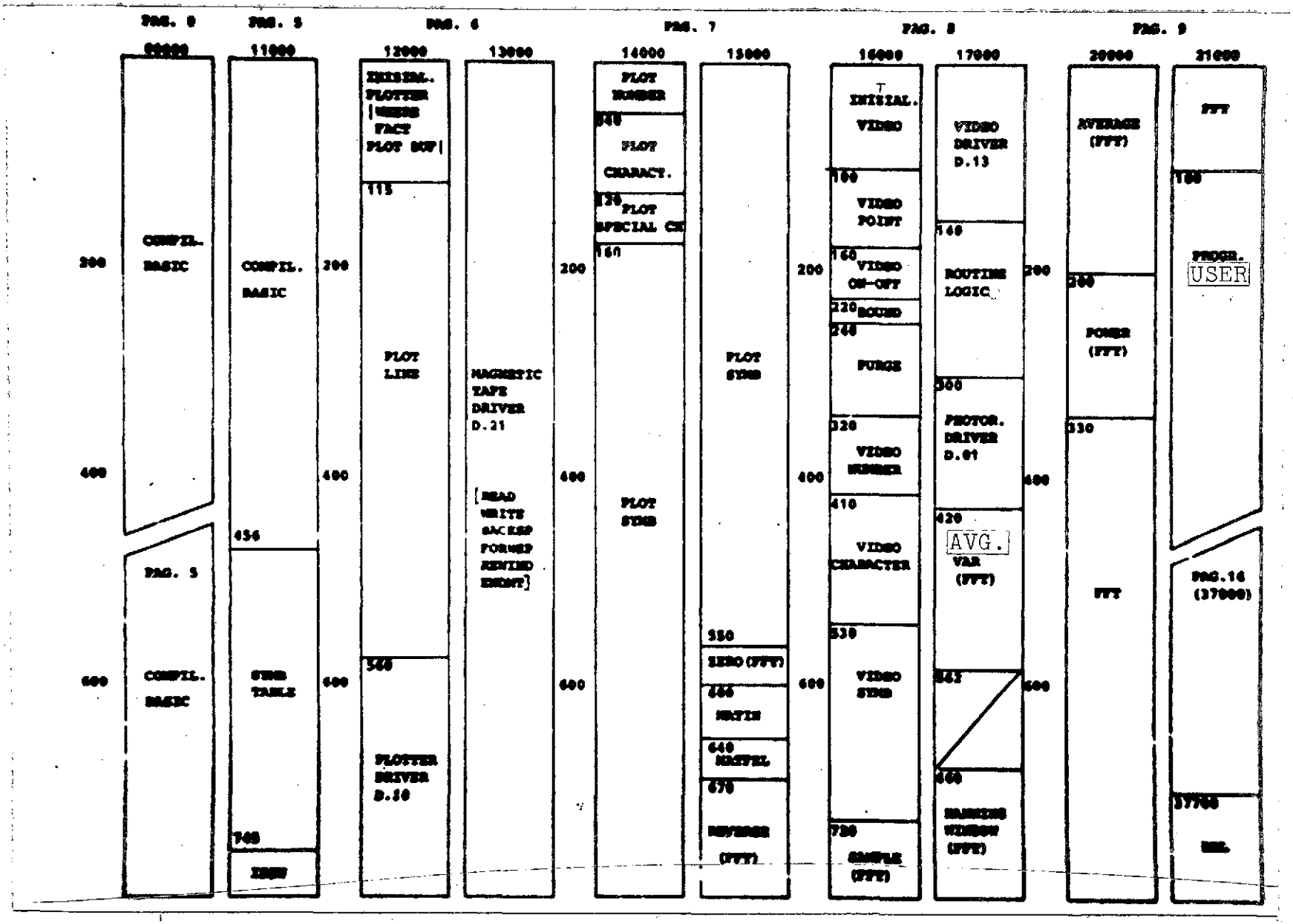



Figure 1. Memory map

program while an input-output operation is carried out. This procedure involves a complication in the programs written by the user, /6
and should be employed only in cases where an effective and substantial time saving is achieved. Some examples of programs using this possibility are shown in appendix B.

The FFT programs have been described in detail in [3].

We remind here that they were written to elaborate on data obtained in real time, or coming from a magnetic or paper tape, following well defined algorithms (Fourier transfers or power spectrum).

As shown in Figure 1, the FFT system and the input-output system do not overlap and, therefore, may be loaded together; however, this operation should be carried out only if required, to avoid decreasing the memory available to the user. The part of the memory used by the input-output system and the FFT is reported in Figure 2.

The program for the acquisition of data from laboratory instrumentation (Laben, voltmeters, and HP counters) have not been included in the input-output system described here, but are available in the Basic version described in [2]; the pertinent calls were reported here. One can observe from Figure 2 that the amount of memory used is lower than for the previous cases.

## 4. Program Calls

The Basic language instruction to call a subprogram is of /7
the type:

```
CALL (N, A, B, C)
```

where N is the identification number of the program (from 1 to 64), and A, B, C are parameters for the transfer of information from the principal program to the subprogram.

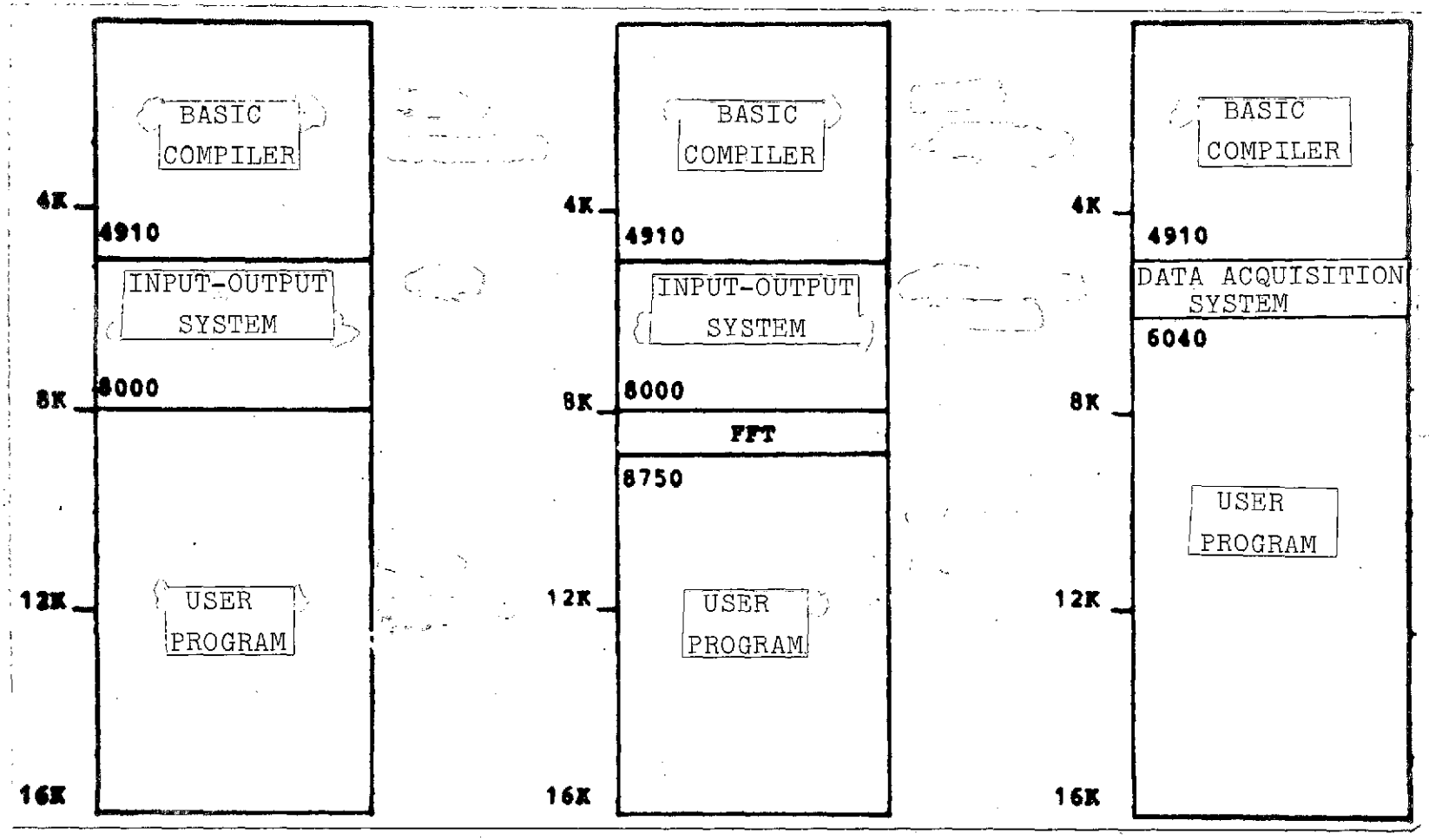



Figure 2. Memory allocation

## 4.I. Input-output system

ISSW: This subprogram controls the position of the $i^{th}$ switch of the "switch register"; it is used mainly to control the machine operations during the execution of the main program.

Call: CALL (1.I.X) - ISSW
I-switch to control ($0 \leq I \leq 15$);
x return datum in floating.

Return: x >= 0 if I OFF (down);
x < 0 if I ON (up)

MAGNETIC TAPE: This group of subprograms permits the transfer of data from the computer to the tape unit, and vice versa, and the control of the operations of the tape unit. Diagnostic messages are inserted which signal on the teletype wrong maneuvers of the tape unit, writing or reading errors, etc. /8

Data are transferred to the tape in binary code; during writing, each word (16 bits) is divided into three characters of 6 bits (2 redundant bits), and the characters are written on the tape without further procedures; during reading, three character groups are regrouped in a 16 bit word.

Read, write, call: CALL (3, A(1), N, I) - READ/WRITE
A(1) or A(1, 1) — first element of the vector (or matrix) of data to be transferred;
N — vector length;
I - = 1 read call;
= 2 write call.

This subprogram transfer a vector of N floating numbers (32 bits) to a record 6*N characters long, or vice versa; the partial reading of a record is possible only if the characters actually read (6*N) do not occupy the whole length of the record. /9

Call for unit control:

CALL (4, 3, N) - BACKSAPCE
jumps back to N record

CALL (4, 4, N) - FORWARDSPACE
jumps forward to N record

CALL (4, 5, 0) - REWIND

CALL (5) - ENDMT
interrupts the program execution until the I/O operation is finished.

PAPER TAPE: This subprogram permits the reading of a paper tape by the fast reader in binary code.

Call: CALL (9, A(1), N) — reads N data (of 16 bits) from the paper tape and transfers them to vector A.
CALL (10) — suspends the execution of the program until the I/O operation of the photoreader is finished.

VIDEO: The driver for the HP1300A Video is composed of some subprograms which permit the display on a screen of graphs, drawings, numbers, and comments in alphanumeric characters; some service programs which carry out the partial or total erasure of graphs and characters on the screen are also included. /10
The video interface card is composed of two 8 bit D/A converters; it converts a 16 bit word

into two analog voltages proportional to the coordinates of the point to be represented on the screen. The resolution on both axes, is of 1/256 (8 bits).

A particular memory zone (output buffer) is filled with the coordinates of the points to be represented on the screen (a memory word per point); the content of the output buffer is transferred to the screen, after conversion, by request of the interface card.

The output of data from the memory to the screen occurs by DMA (direct memory access control); the output time for a point is of approximately five microseconds, and in the time elapsed during two refreshing calls (20 milliseconds) an output of approximately 4,000 points may occur.

Initial call: CALL (11, A(1), N) - INITIAL VIDEO.

This initial subroutine, called at the beginning of the program, establishes the address and length of the data output buffer. /11

A(1) or A(.,1) — first point of the output buffer.

N — buffer length.

Note: DIM A(a) with: a = N/2

Point output call: CALL (12, X, Y) - VIDEO POINT (X, Y)

This subroutine:

— receives from the program the X and Y coordinates of the point to be represented on video;

— transforms X and Y into two integers of 8 bits (maximum video resolution), and puts them in one computer word (of 16 bits);

— places in the output buffer, defined by the initial call, the point (X, Y) which occupies a memory location.

Note: $0 \leq X \leq 255$
$0 \leq Y \leq 255$

Interrupt call: CALL (13, I) - VIDEO ON-OFF

I = 0 — video off

I = 1 — video on

The "refreshing" operation may be inhibited and, therefore, the screen turned off, with CALL (13, 0); the screen may be turned on with CALL (13, 1). The /12\ interrupt system of the machine always remains in operation.

Note: Before calls to peripherals with higher priority than Video (Magnetic Tape, fast reader), which work without DMA, it is necessary to inhibit the Video with CALL (13, 0).

Cancel-lation call: CALL (14, M) - BOUND

CALL (15, M1, M2) - PURGE

The first call gives as output a number M which represents the address of the first three words of the output buffer; the second call cancels the buffer content between the two limits M1 and M2 specified in precedence in subroutine 14.

Call number: CALL (16, X, Y, N) - NUMBER

A variable N is presented on the screen in the Basic standard format with origin at point (X, Y).

Comment call: CALL (17, X, Y, N) - CHARACTER

A comment identified by the REM statement with a number of order N, appears on video with origin (X, Y).

## LOGIC OPERATIONS

These programs permit the number conversion from integer to floating and vice versa, and logical operations with the numbers. /13

Conversion from floating to integer:

CALL (30, F(1), I(1), N) - MATIN
Converts N floating numbers to integers and transfers them from F(K) to I(K).

Note: DIM F(N), I(N/2).

Conversion from integer to floating:

CALL (31, I(1), F(L), N) - MATFL
Converts N integer numbers to floating and transfers them from I(K) to F(K).

Note: DIM F(N), I(N/2).

Logic routine:

CALL (35, 01, 02, K, A, B) - LOGIC
Transfer K bits of word A (32 bits), starting from bit 01, to word B (32 bits), starting from bit 02; K bits are counted in the direction of the most significant bits; the remaining bits of B are left unchanged.

These relationships must be respected:
$0 \leq 01,\ 02 \leq 31$
$1 \leq K \leq 32$.

Note: A 32 bits word is contained into subsequent memory locations, (m) and (m + 1); the less significative 16 bits (0 - 15) are contained in location (m + 1), and the most significative 16 bits (16 - 31) in location (m). /14

Plotter Calcomp

These programs permit the output of data on plotter Calcomp 565 as graphs, and the writing of numbers, comments, symbols, etc.

Calls:

CALL (40, X, Y) - WHERE
Provides in output the coordinates of the pen.

CALL (41, N) - FACT
N-scale factor, in inches; if not called, it is assumed that N = 1.

CALL (42, A(1), N) - PLOTBUF
A — output buffer
N — buffer length
If DIM A(a), then N = 2*a
If this routine is not called, routine PLOT (43) operates on an internal buffer, ten words long; to commute from external to internal buffer, one places N = 0 (for any A).

CALL (43, X, Y, I) - PLOT LINE
Traces a straight line from the initial position of the pen to point (X, Y). /15
I: +2 — Pen down — origin unchanged
+3 — Pen up — origin unchanged
-2 — Pen down — new origin
-3 — Pen up — new origin

CALL (44, X, Y, S, T, N) - PLOT NUMBER
Writes a number N in the standard Basic format.
(X, Y) — number origin
S — dimensions of the alphanumeric characters in inches
T — writing angle, in degrees
N — number in output

CALL (45, X, Y, S, T, N) - PLOT CHARACTER
Writes a comment defined by a REM "statement".
(X, Y) - S - T — as above
N — order number of REM "statement".

CALL (46, X, T, S, T, N) - SPECIAL CHARACTER
Writes special non-alphanumeric characters
(X, Y) - S - T — as above
N — order number of the special character (see list, page 15).

CALL (47) - ENDPLOTTER /16
Stops execution of the program until conclusion of the I/O operations of the plotter.

## List of Diagnostic Messages

During execution of the programs in the input-output system, error causes may occur; for some of these a diagnostic message in code is provided. This has the form: I/O ERROR n.

The execution of the program may be interrupted (STOP) or continued (CONTINUE). The meaning of n is specified in the following table.

01 — Conversion from floating to integer of a number $|N| > 32768$ ($2^{15}$) (routine 30); STOP.

12 — Magnetic tape: command to write with protected tape; STOP.

13 — Magnetic tape: command to read with tape in EOT; STOP.

14 — Magnetic tape: "Timing error"; CONTINUE.

15 — Magnetic tape: "Parity error", CONTINUE.

16 — Magnetic tape: routine call 3 wrong; I different from 1 or 2; STOP. /17

17 — Magnetic tape: routine call 4 wrong; control parameter different from permitted values (3, 4, or 5).

TABLE OF PLOTTER SPECIAL CHARACTERS
(SUBROUTINE 46)

| N | Character | N | Character |
|---|---|---|---|
| 1 | ⊙ | 14 | * |
| 2 | △ | 15 | - |
| 3 | + | 16 | \| |
| 4 | × | 17 | ↓ |
| 5 | ◇ | 18 | ≤ |
| 6 | ♠ | 19 | ≥ |
| 7 | ⋈ | 20 | ¬ |
| 8 | Z | 21 | ∟ |
| 9 | Y | 22 | ≠ |
| 10 | ⌑ | 23 | ° |
| 11 | ✱ | 24 | ▪ |
| 12 | ⧖ | 25 | ± |
| 13 | ' | | |

## 4.2. FFT programs

CALL (50, N, L, K, B, D (1, 1), E(1, 1), W(1), S) - FFT
Carries out a Fourier transfer of N data, contained in /18
the matrix D; the result is contained in D (real part) and in E (complex part).

N = $2^L$ — number of samples ($\leq$ 1024);
K — = 0 direct transfer;
= 1 inverse transfer;
B = 12 — number of bits of the weight (W) function;
D(1, 1) — data matrix;
E(1, 1) — data matrix;
DIM D(a, b), E(a, b)
with a*b = N/2;
W(1) — vector of the weight function cos (2πn/N)
DIM W(a), with a = N/8;
S — scale factor.

CALL (51, N, C(1, 1), A(1, 1), R(1, 1), J, S1) - AVERAGE
Carries out the overall average of spectra
N — sample number;
C(1, 1) — matrix containing the spectra average;
DIM C(a, b) with a*b = N/4;
A(1, 1) — matrix containing the single spectrum;
DIM A(a, b) with a*b = N/2;
R(1, 1) — matrix of inverse indices; /19
DIM (a, b) with a*b = N/2;
J — number of calculation cycles;
S1 — scale factor.

CALL (52, N, A(1, 1)) - ZERO
Makes equal to zero matrix A, N words long
DIM A(a, b) with a*b = N/2.

CALL (53, N, D(1, 1), W(1)) - HANNING WINDOW
Applies the Hanning window algorithm on the data contained in D. The meaning of the parameters is the same that was already specified.

CALL (54, N, R(1, 1)) - REVERSE
Calculates inverse indices and places them in the R memory.

CALL (55, N, D(1, 1) - SAMPLE
Acquires N data and places them in the D memory. Data have a 9 bit format (8 bits + sign), integer, completment of 2.

CALL (56, N, D(1, 1), E(1, 1), A(1, 1)) - POWER
Carries out the calculation of the amplitude spectrum and places in the A memory the results, starting from the real and imaginary part of the Fourier transfer, contained respectively in D and E. /20

CALL (57, N, D(1, 1), M, V) - AVERAGE - VAR
Carries out the calculation of the average and of the variance of the data acquired by routine 55 and eliminates the continuous component. The average and variance values are provided in output, respectively, in M and V.

## 4.3. Data acquisition system

The data acquisition system is described in report [2]; we reproduce here the program calls contained in the report, and the descriptions of the programs not listed in the previous paragraphs.

### ISSW

CALL (1, I, X) - ISSW

### LABEN

CALL (2, X)
This program carries out the data transfer from the Laben multichannel, connected to the I/o n.15 channel,

to the computer; it also carries out the conversion form BCD to binary floating. X is the return datum.

MAGNETIC TAPE

CALL (3, A(1), N) - WRITE
CALL (4, A(1), N) - READ
CALL (5) - REWIND /21
CALL (6, N) - FORWARD SPACE
CALL (7, N) - BACKSPACE

FAST READER

CALL (8, A(1), N) - PUNCH
CALL (9, A(1), N) - READ

VIDEO

CALL (11, A(1), N) - SET PLOT (INITIAL)
CALL (12, X, Y) - PLOT POINT
CALL (13, I) - VIDEO ON-OFF
CALL (14, M) - BOUND
CALL (15, M1, M2) - PURGE
CALL (16, X, Y, N) - NUMBER
CALL (17, X, Y, N) - CHARACTER
CALL (18) - CAMERA
CALL (19) - SINGLE REFRESHING

DATA ACQUISITION

CALL (20, X) - VOLT 1
X — voltmeter reading; channel 16

CALL (21, X) - VOLT 2
X — voltmeter reading; channel 20

CALL (22, X) - COUNT
X — counter reading; channel 17

CALL (23, X) - INPUT 1
X — input datum (16 bits); channel 21

CALL (24, X) - INPUT 2 /22
X — input datum (16 bits); channel 22

CALL (25, X) - OUTPUT 1
X — output datum (16 bits); channel 21

CALL (26, X) - OUTPUT 2
X — output datum (16 bits); channel 22

CALL (27, X) - OUTPUT 3
X — output datum (16 bits); channel 21.
Without commands ENCODE AND FLAG.

LOGIC OPERATIONS

CALL (30, F(1), I(1), N) - MATIN
CALL (31, I(1), F(1), N) - MATFL

# APPENDIX A

## A. Operations with BBL (Basic Binary Loader)

The BBL is a program capable of loading in the memory absolute programs (Fortran, Assembler, Basic, Editor, DCS); it is permanently placed in the last 64 positions of the memory, and is protected by a specific switch (LOADER SWITCH) on the computer panel.

There are two versions of BBL, one for loading using magnetic tape (Magnetic BBL) and one for loading using paper tape (BBL Standard).

### A1. Loading of an absolute program with BBL from magnetic tape; MAGNETIC BLL.

1. Turn computer on (POWER ON).
2. Turn magnetic tape on.
3. Place SWITCH REGISTER in position 037700 (starting address of BBL).
4. Press LOAD ADDRESS.
5. Load in SWITCH REG. the identification number of the program to be loaded (to be read on the tag attached to the tape).
6. Press PRESET.
7. Put LOADER SWITCH in ENABLED.
8. Press RUN.
9. The computer should stop with the number 102077 in the T register (the first on the top). If there is a different number, some error was made. Try again from point 3. /24
10. Place LOADER SWITCH in PROTECTED.

### A2. Loading of an absolute program with BBL from a photoreader (BBL Standard).

1. Turn computer on.
2. Turn photoreader on.

3. Place SWITCH REG. in position 037700.
4. Press LOAD ADDRESS.
5. Place in the photoreader the tape of the program to be loaded.
6. Press PRESET.
7. Place LOADER SWITCH in ENABLED.
8. Press RUN.
9. Check that computer stopped with the number 102077 in the T register.
10. Place LOADER SWITCH in PROTECTED.

A3. Procedures to replace the magnetic BBL with a standard BBL or vice versa.

These procedures have to be applied when one changes from a standard system (paper tape) to the magnetic system or vice versa. Normally a magnetic BBL is placed in the memory.

— From standard BBL to magnetic BBL

1. Load with the BBL (procedure A2) the tape "EXCHANGE PAPER FOR MAG. TAPE".
2. Place SWITCH REG. in position 000002.
3. Press LOAD ADDRESS. /25
4. Place LOADER SWITCH in ENABLED.
5. Press RUN.
6. Place LOADER SWITCH in PROTECTED.

— From magnetic BBL to standard BBL

1. Load with BBL (procedure A1) the EXCHANGE program from the magnetic tape.
2. Place LOADER SWITCH in ENABLED.
3. Press RUN.
4. Place LOADER SWITCH in PROTECTED.

A4. Loading of magnetic BBL.

This permits the replacement of the BBL in the memory in the case of its total destruction.

1. Load number 000020 in SWITCH REG.
2. Press LOAD ADDRESS.
3. Load in the memory the following 12 instructions, placing in SWITCH REG. the instruction code, and pressing LOAD MEMORY. The address, contained in register M, advances automatically in synchronism with the command LOAD MEMORY.

| ADDRESS | CODE | INSTRUCTION |
|---|---|---|
| 20 | 103712 | STC, CLF 12 |
| 21 | 102312 | SFS 12 |
| 22 | 024021 | JMP 21 |
| 23 | 102512 | LIA 12 |
| 24 | 001727 | ALF, ALF |
| 25 | 103712 | STC, CLF 12 |
| 26 | 102312 | SFS 12 |
| 27 | 024026 | JMP 26 /26 |
| 30 | 102412 | MIA 12 |
| 31 | 170001 | STA B, I |
| 32 | 006004 | INB |
| 33 | 024020 | JMP 20 |

4. Place the BBL MAG. tape in the photoreader.
5. Load in SWITCH REG. 037700 and press LOAD B.
6. Load in SWITCH REG. 000020 and press LOAD ADDRESS.
7. Place LOADER SWITCH in ENABLED.
8. Press PRESET, then RUN.
9. The computer does not stop automatically. Press HALT.
10. Place LOADER SWITCH in PROTECTED.

B. Operations with Basic.

B1. Loading of BASIC

1. Load BASIC in the memory using BBL (procedures A1 and A2). If BASIC is already in the memory, go directly to point 2.
2. Load number 000100 in SWITCH REG.
3. Press LOAD ADDRESS.
4. Press PRESET, then RUN.
5. The TTY has to write READY.
6. The program may be written on the TTY or loaded from tape, according to procedure B2.

B2. Loading of the program from paper tape. /27

1. Place the tape on the photoreader.
2. Write on the TTY the word PTAPE, followed by "return". The tape is read automatically.

B3. Output of the program on paper tape.

1. Turn ON the puncher on the TTY.
2. Write on the TTY the word PLIST, followed by "return".

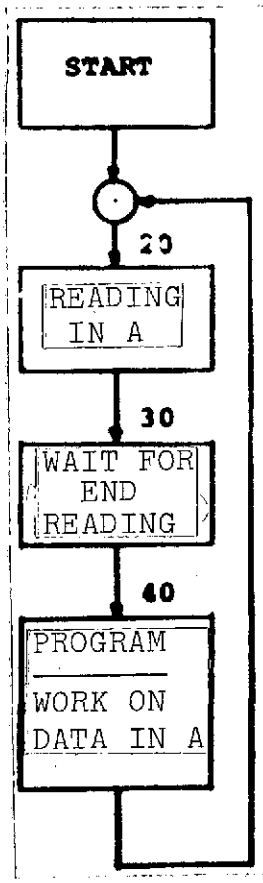
B4. Turning off the computer.

1. Place computer in HALT.
2. Turn off peripherals (TTY, photoreader, magnetic tape, etc.).
3. Turn computer off.

APPENDIX B

Typical programs from the reading of the magnetic tape using the HP2116B. /28

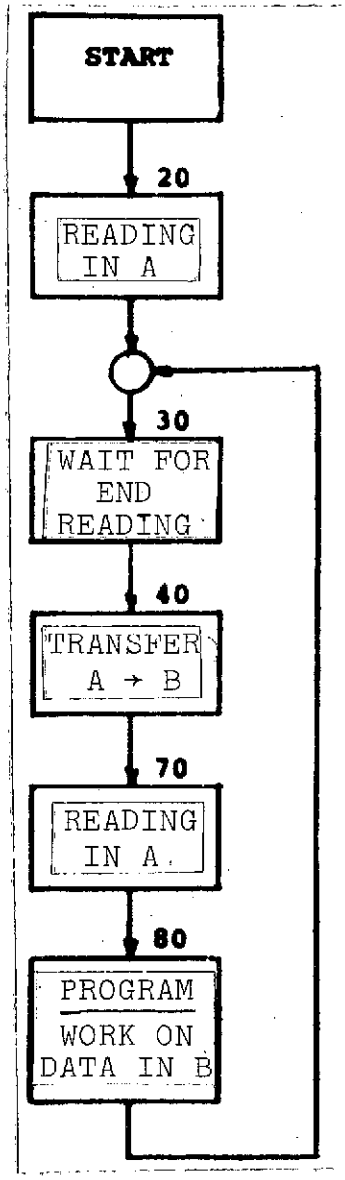
1. Tape reading with single buffer A; no overlap between reading and calculation.




```
10  DIM A(N)
20  CALL (3,A(1),N,1)
30  CALL (5)
40  PROGRAM
      :
 N  GO TO 20
```

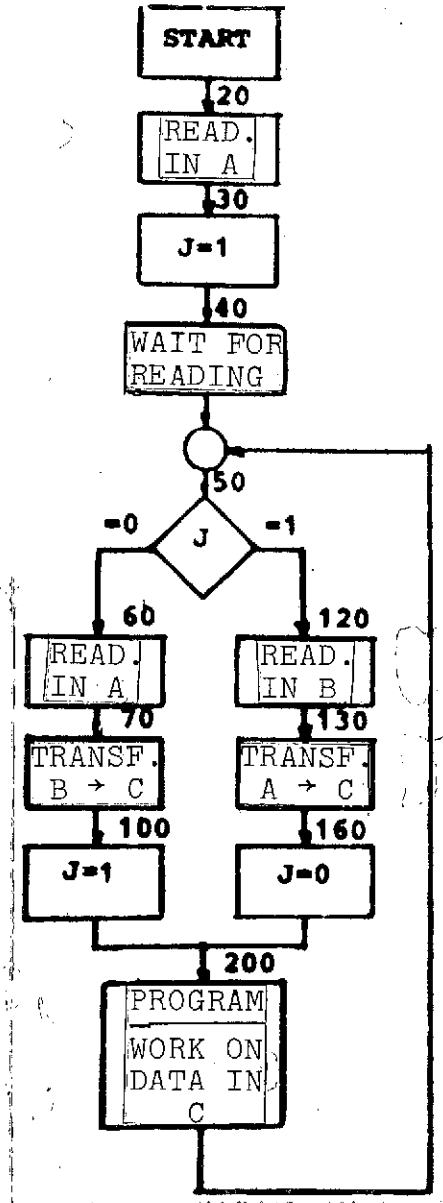
2. Tape reading with single buffer A, and transfer to B; partial overlap of reading and calculation.

/29




```
10   DIM A(N), B(N)
20   CALL (3,A(1),N,1)
30   CALL (5)
40   FOR I=1 TON
50   LET B(I) = A(I)
60   NEXT I
70   CALL (3,A(1),N,1)
80   PROGRAM
       .
       .
       .
 N   GO TO 30
```

3. Tape reading with double buffer A and B, and transfer in C. /30
Optimum overlap of reading and calculation.




```
 10  DIM A(N),B(N),C(N)
 20  CALL (3,A(1),N,1)
 30  LET J=1
 40  CALL (5)
 50  IF J>0 THEN 120
 60  CALL (3,A(1),N,1)
 70  FOR I=1 TO N
 80  LET C(I) = B(I)
 90  NEXT I
100  LET J=1
110  GO TO 200
120  CALL (3,B(1),N,1)
130  FOR I=1 TON
140  LET C(I) = A(I)
150  NEXT I
160  LET J=0
200  PROGRAM
       :
       :
  N  GO TO 50
```

## REFERENCES


1. Palutan, F. Special Programs and Instruction for Use of the HP2116B Computer of the Space Plasma Laboratory. LPS-70-7, October, 1970. /31

2. Palutan, F. Data Acquisition System with the HP2116B Computer LPS-71-23, September, 1971.

3. Palutan, F. System for Signal Harmonic Analysis Using the Computer HP2116B. LPS-72-17, July, 1972.



Translated for National Aeronautics and Space Administration under contract No. NASw 2483, by SCITRAN, P. O. Box 5456, Santa Barbara, California, 93108